

```

#include <iostream>
#include <vector>
using namespace std;

bool isSafe(vector<vector<int>>& board, int row, int col, int n) {
    for (int i = 0; i < row; i++) {
        if (board[i][col] == 1) return false;
    }

    for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) {
        if (board[i][j] == 1) return false;
    }

    for (int i = row, j = col; i >= 0 && j < n; i--, j++) {
        if (board[i][j] == 1) return false;
    }

    return true;
}

bool solveNQueensUtil(vector<vector<int>>& board, int row, int n) {

    if (row == n) return true;

    for (int col = 0; col < n; col++) {
        if (board[row][col] == 1) {

            return solveNQueensUtil(board, row + 1, n);
        }
    }

    for (int col = 0; col < n; col++) {
        if (isSafe(board, row, col, n)) {
            board[row][col] = 1;

            if (solveNQueensUtil(board, row + 1, n)) {
                return true;
            }

            board[row][col] = 0;
        }
    }
}

```

```

    return false;
}

void printBoard(const vector<vector<int>>& board, int n) {
    cout << "\nFinal n-Queens matrix:\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << board[i][j] << " ";
        }
        cout << endl;
    }
}

int main() {
    int n;
    cout << "Enter the size of the board (n): ";
    cin >> n;

    vector<vector<int>> board(n, vector<int>(n, 0));

    int firstRow, firstCol;
    cout << "Enter the row and column of the first queen (0-indexed): ";
    cin >> firstRow >> firstCol;

    if (firstRow < 0 || firstRow >= n || firstCol < 0 || firstCol >= n) {
        cout << "Invalid position! Please enter values between 0 and " << n-1 << endl;
        return 1;
    }

    board[firstRow][firstCol] = 1;

    cout << "\nFirst queen placed at position (" << firstRow << ", " << firstCol << ")\n";

    if (solveNQueensUtil(board, 0, n)) {
        printBoard(board, n);
        cout << "\nSolution found successfully!" << endl;
    } else {
        cout << "\nNo solution exists with the first queen placed at position ("
            << firstRow << ", " << firstCol << ")" << endl;
    }

    return 0;
}

```

**OUTPUT:**

Enter the size of the board (n): 4

Enter the row and column of the first queen (0-indexed): 1

0

First queen placed at position (1, 0)

Final n-Queens matrix:

1 0 0 0

1 0 0 0

0 0 0 1

0 1 0 0

Solution found successfully!